Distributed Optimization of Complex Simulation-Based Systems

Natalia M. Alexandrov

Multidisciplinary Optimization Branch
NASA Langley Research Center
Hampton, Virginia

http://mdob.larc.nasa.gov

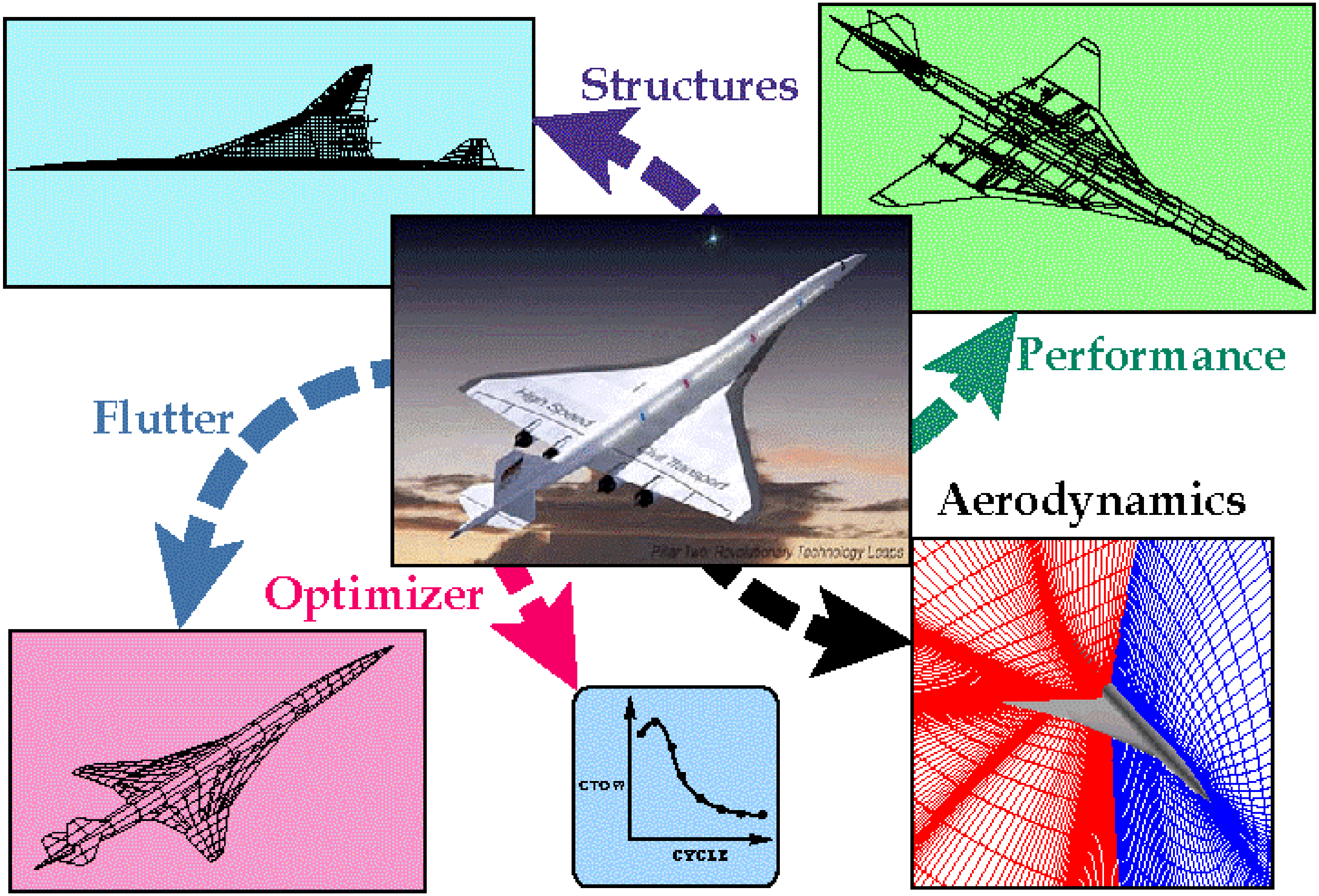
n.alexandrov@larc.nasa.gov

Outline

- A brief introduction to MDO
- MDO problem synthesis
  - Background
  - Analysis of a promising MDO approach
  - An alternative approach
- Concluding remarks

Courtesy J.A. Samareh

High-fidelity design of an aerospace vehicle

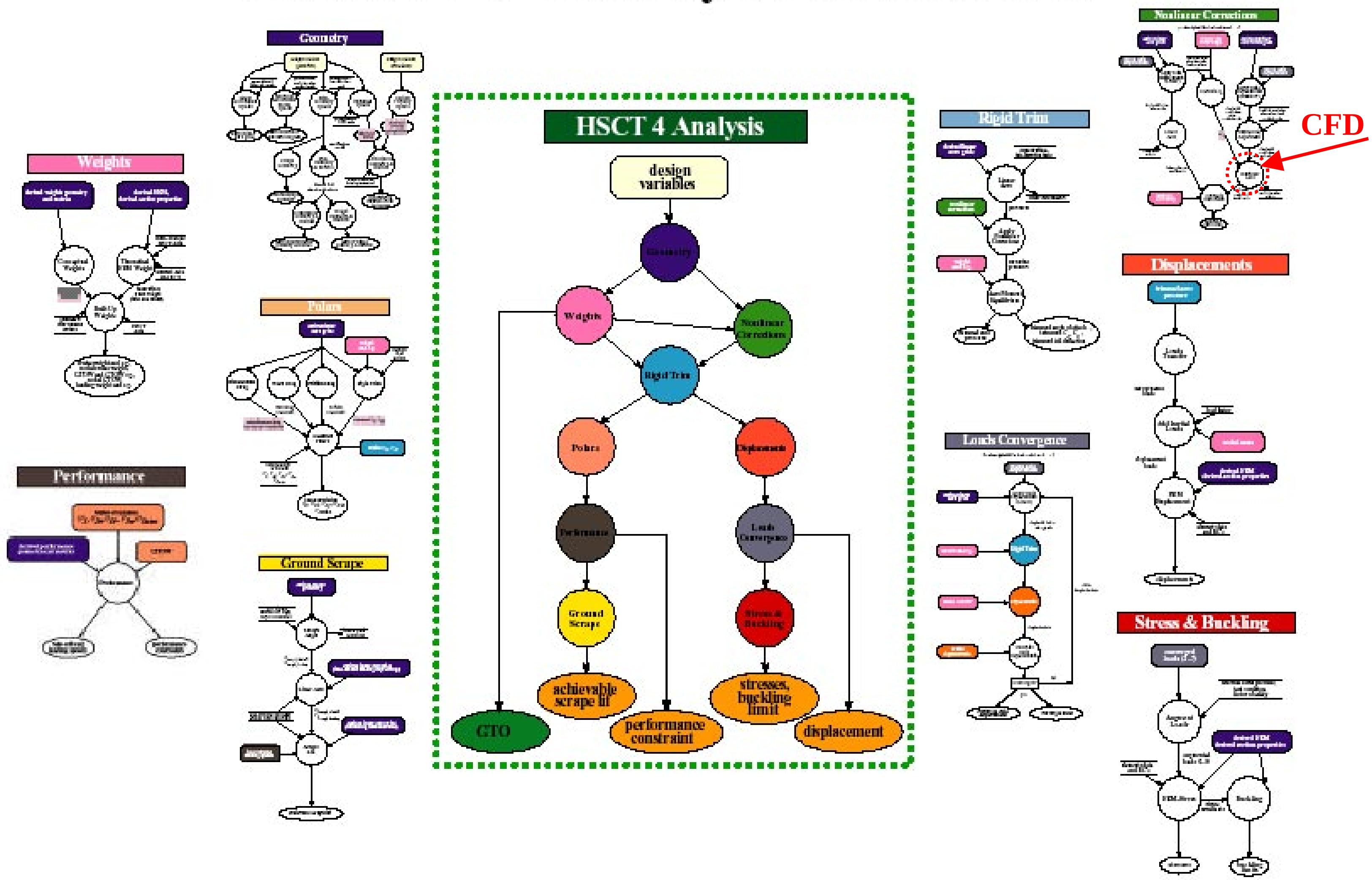


Multidisciplinary Design Optimization (MDO)

- Systematic approaches to the design of complex, coupled systems
- “Multidisciplinary” – different aspects of the design problem
- For now: MDO is the subset of the total design problem that can be expressed as an NLP
- MDO involves many areas
  - Design-oriented analysis
  - Design problem synthesis and solution
  - Computational infrastructure

Courtesy J.A. Samareh

Full HSCT 4 Analysis Procedures

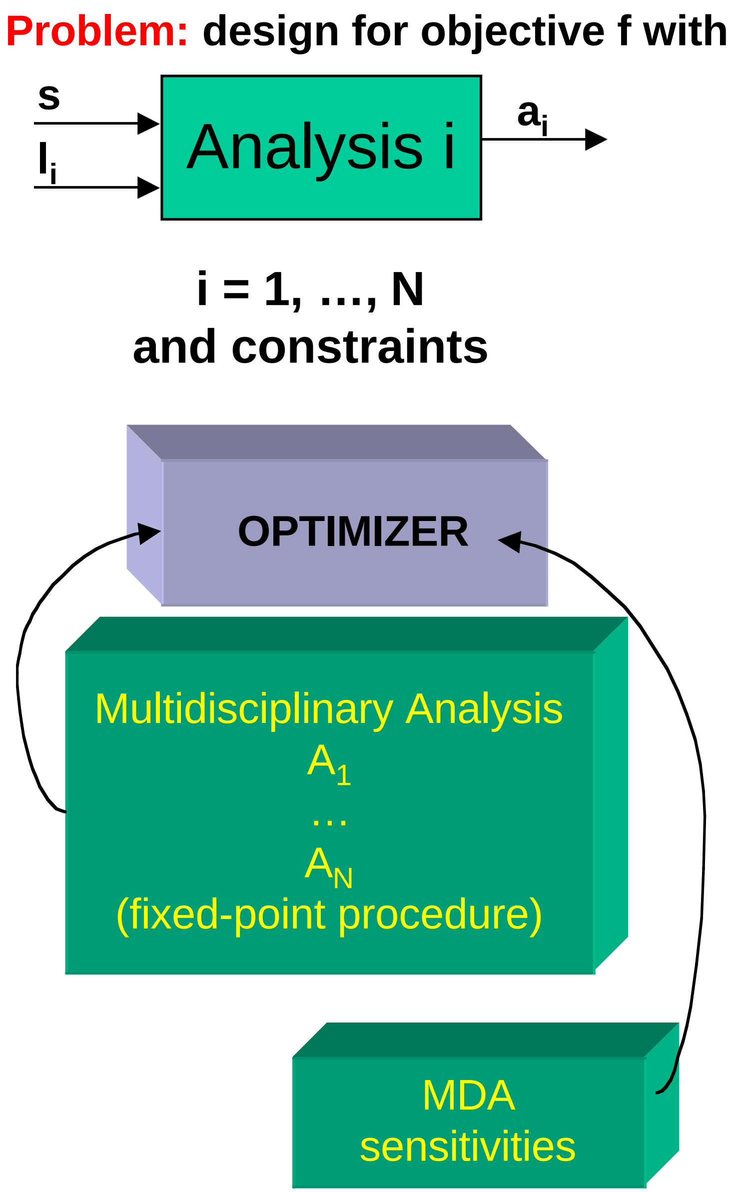


n.alexandrov@larc.nasa.gov

A component of MDO: problem synthesis

- MDO problem formulation
  – Relatively recent (e.g., Cramer et al., 1992) area that deals with stating the MDO problem as an NLP
- Analytical features of MDO problem formulation strongly influence the practical ability of optimization algorithms to solve the MDO problem reliably and efficiently

Canonical problem synthesis: Fully Integrated Formulation (FIO)



- Laborious, expensive, one-time process
- Inflexible
- Assumes that MDA is done via fixed-point iteration
- Need to develop Multidisciplinary Analysis (MDA) based derivatives
- Expensive to maintain MDA far from solution
- Disciplinary autonomy minimal
- Drawbacks of FIO motivate other formulations

n.alexandrov@larc.nasa.gov

Would like to have…

- A formulation that is easy to implement
- Maximum disciplinary autonomy
  - Letting disciplinary experts design virtually independently (e.g., optimize with respect to local variables and local objectives and constraints in disciplinary subproblems)
- Efficiency in function evaluations
- Good convergence properties
- Flexible re-formulation and hybrids
- Etc.

Some observations…

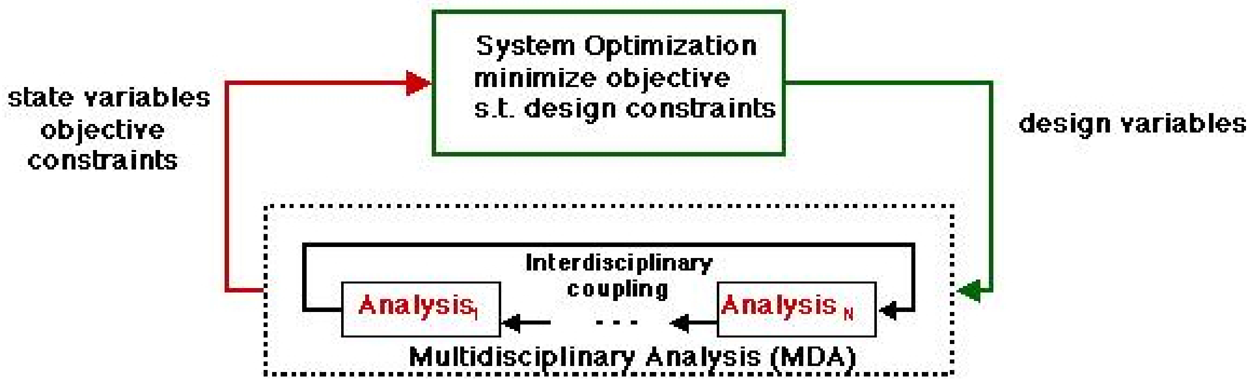
- Many alternatives to FIO are based on *ad hoc* approaches
- Anecdotal evidence indicates that some methods work dramatically better than others
- Much “fine-tuning” goes into solution
- Limited computational evidence of relative performance properties
- Virtually impossible to replicate results
- Now a more systematic analysis in progress

Example: HPCCP/HSCT formulation study

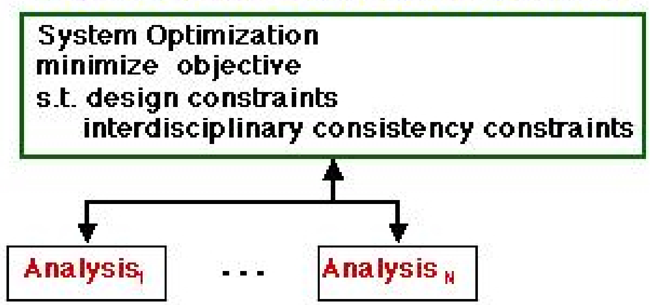
Alexandrov and Kodiyalam, AIAA-98-4884

Evaluated three formulations with respect to several performance metrics:

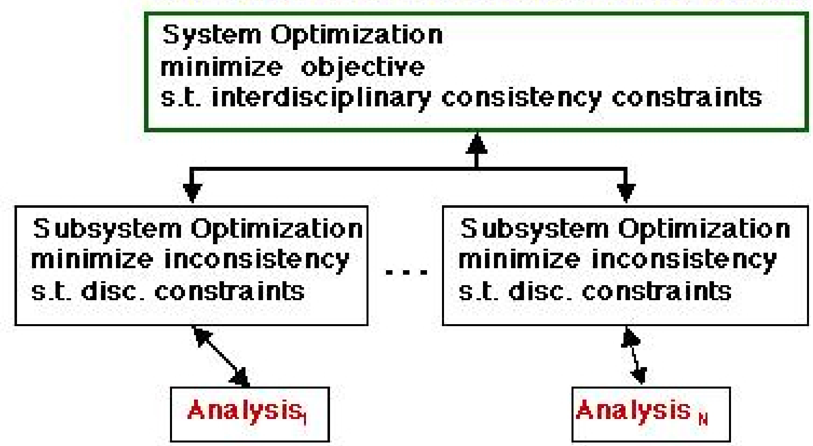
Basic formulation



Equivalent to basic formulation



Not equivalent to basic formulation



n.alexandrov@larc.nasa.gov

Evaluating a formulation

- Amenable to solution?
- Robust?
  - Relationship of the solution set to that of the canonical problem
  - Optimality conditions
  - Sensitivity to perturbations
- Efficient?
- Autonomy of implementation / ease of transformation?
  - The most labor-intensive part
  - Important because no single formulation is good for all problems
- Autonomy of execution?
  - Wish to follow organizational structure for design
  - Wish to optimize wrt local variables only in disciplines
- Direct influence on solubility and software

Example, continued

- Contributing formulations
  - Basic formulation (FIO)
  - Equivalent (Distributed Analysis Optimization, DAO)
  - Non-equivalent (Collaborative Optimization, CO)
- Dramatic differences in performance

Problem Method	1	2	3	4	5	6	7	8	9	10
FIO	**610**	**220**	**610**	**81**	**3234**	**5024**	**8730**	**245**	**1574**	**1353**
CO	**15626**	**19872**	**1785**	**2102**	**837**	**40125**	**691058**	**-**	**-**	**-**
DAO	**9530**	**8976**	**382**	**-**	**544**	**932**	**-**	**-**	**-**	**-**

Example: representative # analyses

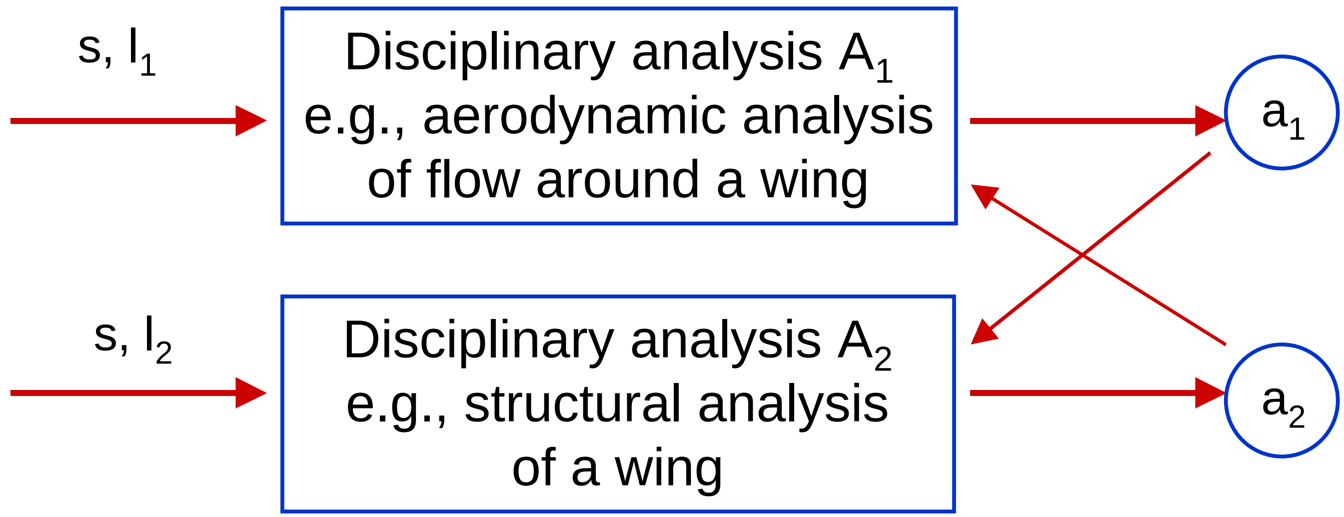
n.alexandrov@larc.nasa.gov

The remainder of the discussion is based on the following publications:

1. Alexandrov, N. M. and Lewis, R. M.: Analytical and Computational Properties of Collaborative Optimization for Multidisciplinary Design, AIAA Journal, 2002, Vol. 40, No. 2, pp. 301-309.
2. Alexandrov, N. M.: Multilevel Methods for Optimal Design, *Encyclopedia of Optimization*, Floudas, C. A. and Pardalos P. M., Eds., pp. 528-537, Kluwer Academic Publishers, 2001.
3. Alexandrov, N. M., Lewis, R. M.: Algorithmic Perspectives on Problem Formulations in MDO, AIAA paper 2000-4719, September 2000.
4. Alexandrov, N. M., Lewis, R. M.: Analytical and Computational Properties of Distributed Approaches to MDO, AIAA 2000-4718, September 2000.
5. Alexandrov, N. M., Lewis, R. M. : Analytical and Computational Aspects of Collaborative Optimization, NASA/TM-2000-210104, April 2000.
6. Alexandrov, N. M.; Lewis R. M.: Comparative Properties of Collaborative Optimization and Other Approaches to MDO, *Engineering Design Optimization*, V.V. Toropov Ed., MCB Press, 1999.
7. Alexandrov, N. M.: Optimization Algorithms in MDO in *Multidisciplinary Design Optimization: State of the Art*, Alexandrov, N. M. and Hussaini, M. Y., pp. 79-89, SIAM Publications, Philadelphia, PA, February 1997.

n.alexandrov@larc.nasa.gov

Two-discipline model problem and some formulations



Multidisciplinary analysis (MDA) expresses the physical requirement that a solution must satisfy both analyses: given (s, l_1, l_2), solve the system

$$a_1 = A_1\,(s, l_1, a_2)$$
$$a_2 = A_2\,(s, l_2, a_1)$$

MDA defines a_1 and a_2 implicitly as functions of (s, l_1, l_2):

$$a_1 = a_1(s, l_1, l_2)$$
$$a_2 = a_2(s, l_1, l_2)$$

n.alexandrov@larc.nasa.gov

Collaborative Optimization (CO)

- Alexandrov and Lewis, Analytical and computational properties of collaborative optimization, AIAA Journal, Feb. 2002, ICASE report (+ related papers)
- CO-like methods have been re-invented or re-discovered every few years for the last 20 years or so; last version due to Kroo et al.
- CO attempts to state and solve MDO problems in a way that preserves the autonomy of disciplinary computations
- An intuitive and attractive approach that appears to mimic the actual design process
- Instructive because of the intrinsic computational difficulties
- A good example of the effect of autonomy on efficiency and robustness

CO (description)

System problem: $\underset{s,\, t_1,\, t_2}{\textbf{minimize}} \quad f(s, t_1, t_2)$

$\textbf{subject to } C(s, t_1, t_2) = 0$

The system problem issues design targets (s, t_1, t_2) to disciplines.
In lower-level problems, the disciplines design to match targets:

In discipline i, given (s, t_i, t_j), compute $\bar{\sigma}_i(s,t_i,t_j)$ and $\bar{l}_i(s,t_i,t_j)$ as solution of the following problem:

$$\underset{\sigma_i, l_i}{\textbf{minimize}} \quad \_[\|\sigma_i - s\|^2 + \|a_i(\sigma_i, l_i, t_j) - t_i\|^2$$

$$\textbf{subject to } g_i(\sigma_i, l_i, a_i(\sigma_i, l_i, t_j)) \le 0,$$

where a_i is computed via the disciplinary analysis $a_i = A_i(\sigma_i, l_i, t_j)$

One form of consistency constraints is

$$c_i(s,t_1,t_2) = \_[\|\bar{\sigma}_i(s,t_1,t_2) - s\|^2 + \|a_i(\bar{\sigma}_i(s,t_1,t_2), \bar{l}_i(s,t_1,t_2), t_j) - t_j\|^2$$

Illustration: World's simplest problem

(e.g., a bar of fixed length and variable cross-section area under a longitudinal force)

$$\text{minimize}\{s \mid 0 \le s \le 1\}$$

On reformulating as CO_2, system and subsystem problems become

$$\begin{aligned} \underset{s}{\text{minimize}} \quad & f(s) \\ \text{subject to} \quad & c_1(s) = \tfrac{1}{2} \parallel s - \sigma_1(s) \parallel^2 = 0 \\ & c_2(s) = \tfrac{1}{2} \parallel s - \sigma_2(s) \parallel^2 = 0 \end{aligned}$$

$$\min\{\frac{1}{2} \parallel \sigma_1 - s \parallel^2 \ \mid \sigma_1 \ge 0\} \ \text{ and } \ \min\{\frac{1}{2} \parallel \sigma_2 - s \parallel^2 \ \mid \sigma_2 \le 1\}$$

One readily checks that the subproblem solutions are

$$\sigma_1(s) = \begin{cases} 0 & \text{if } s \le 0 \\ s & \text{if } s \ge 0 \end{cases} \qquad \sigma_2(s) = \begin{cases} s & \text{if } s \le 1 \\ 1 & \text{if } s \ge 1 \end{cases}$$

Example continued

Breakdown of the standard stationarity conditions in CO_2

- $\nabla c_i(s) = s - \sigma_i(s)$ and at $s_* = \alpha, \nabla c_1(s_*) = 0$
- Stationarity conditions: there exist λ_1 and λ_2 such that

$$\nabla f(s_*) + \lambda_1 \nabla c_1(s_*) + \lambda_2 \nabla c_2(s_*) = 0$$

- But $\nabla f(s_*) + \lambda_1 \nabla c_1(s_*) + \lambda_2 \nabla c_2(s_*) = \nabla f(s_*) = 1$

Computational difficulties occur near solutions. E.g., could start at a solution and not recognize it.

Example continued:

Results of NPSOL with $s_0 = 0.001$ and $s_* = 0$

Iteration	s	Penalty
0	1.000e-03	0.0e+00
1	-9.990e-01	4.2e+00
2	-9.847e-01	5.7e+00
3	-8.282e-01	7.4e+00
4	-4.142e-01	2.7e+01
5	-3.430e-01	5.9e+01
6	-1.718e-01	4.0e+02
7	-1.436e-01	8.2e+02
8	-7.251e-02	5.4e+03
9	-6.076e-02	1.1e+04
10	-3.203e-02	6.5e+04
11	-2.717e-02	1.2e+05
12	-1.727e-02	5.1e+05
13	-1.442e-02	1.9e+06
14	-1.414e-02	4.7e+06

Some properties of CO of algorithmic import

- No need for MDA until solution
- Local variables handled in disciplines
- No hope for large bandwidth of coupling
- System-level problem is more nonlinear than the original
- Jacobian of the system-level constraints vanishes at every feasible point of the system-level problem $\Rightarrow$ Lagrange multipliers will not exist, in general for the system-level problem
- Difficulties occur at or near points of interest (multidisciplinary feasible)
- Attempts to relax the problem lead to unpredictable results
- Difficulties due to reformulation even if the original problem is perfectly well behaved

. . .

More observations

- Other distributed optimization methods have been proposed and all suffer from similar difficulties: coupling must be resolved
- Eliminating local variables via optimization problems may cause difficulties
- Conjecture: for broadly and/or strongly coupled MDO problems, disciplinary autonomy of calculations is at odds with computational robustness and efficiency
- Perhaps, can sacrifice some measure of autonomy for robustness and efficiency
- Distribute computation via more conventional optimization formulations and attendant algorithms

Alternatives formulations

- Start with a simultaneous analysis-and-design formulation (SAND or AAO)
- SAND is related to several other formulations via constraint closure
- Gradients for SAND, FIO, and DAO (including in-between formulations) are related
- Start with an algorithm for SAND and arrive at algorithms for FIO and DAO via simple modifications that involve closing specific sets of constraints

Relationship among Optimization Problem Formulations

Write MDA as
$$\begin{aligned} a_1 &= A_1(s, l_1, t_2) \\ a_2 &= A_2(s, l_2, t_1) \\ t_1 &= a_1 \\ t_2 &= a_2 \end{aligned}$$

Start with Simultaneous Analysis and Design (SAND) formulation:

$$\begin{array}{ll} \underset{s, a_1, a_2, l_1, l_2, t_1, t_2}{\text{minimize}} & f_{SAND}(s, a_1, a_2) \\ \text{subject to} & g_1(s, l_1, a_1) \geq 0 \\ & g_2(s, l_2, a_2) \geq 0 \\ & a_1 = A_1(s, l_1, t_2) \\ & a_2 = A_2(s, l_2, t_1) \\ & t_1 = a_1 \\ & t_2 = a_2 \end{array}$$

n.alexandrov@larc.nasa.gov

Relationship among Optimization Problem Formulations (cont.)

- **Eliminate subsets of variables from SAND by *closing* various subsets of constraints $\Longrightarrow$ get other formulations:**
  - **Distributed Analysis Optimization (DAO):** Eliminate a_1, a_2 as independent variables by closing the disciplinary analysis constraints at every iteration of optimization
  - **Fully Integrated Optimization (FIO):** In addition, eliminate t_1, t_2 as independent variables by closing $t_1 = a_1$ and $t_2 = a_2$.
  - **Optimization by Linear Decomposition (OLD):** Eliminate l_1, l_2, t_1, t_2 as independent variables via optimization subproblems (MDA remains)
  - **Collaborative Optimization (CO):** Eliminate l_1, l_2 (but not t_1, t_2) via optimization subproblems

Autonomy / modularity in implementation

- Computational elements needed for optimization (in particular, sensitivities) can be implemented autonomously by disciplines
- All formulations require roughly the same amount of work to implement
- Consider sensitivities…

Example: Sensitivities in DAO vs FIO

Consider DAO:

$$
\begin{array}{ll}
\underset{s,l_1,l_2,t_1,t_2}{\textbf{minimize}} & f_{DAO}(s,t_1,t_2) = f(s, a_1(s,l_1,l_2,t_2), a_2(s,l_1,l_2,t_1)) \\
\textbf{subject to} & g_0(s,t_1,t_2) \geq 0 \\
& g_1(s,l_1,t_1) \geq 0 \\
& g_2(s,l_2,t_2) \geq 0 \\
& t_1 = a_1(s,l_1,l_2,t_2) \\
& t_2 = a_2(s,l_2,l_2,t_1),
\end{array}
$$

where, given $(s,l_1,l_2,t_1,t_2), a_1$ and a_2 are found from

$$
\begin{aligned}
a_1 - A_1(s,l_1,t_2) &= 0 \\
a_2 - A_2(s,l_2,t_1) &= 0.
\end{aligned}
$$

n.alexandrov@larc.nasa.gov

Example: Sensitivities in DAO vs FIO, cont.

For the objective $f_{DAO}(s, t_1, t_2)$, we need

$$\frac{\partial f}{\partial s}, \frac{\partial f}{\partial t_1}, \frac{\partial f}{\partial t_1}$$

For the design constraints $g_1(s, l_1, t_1)$ and $g_2(s, l_2, t_2)$ we need

$$\frac{\partial g_1}{\partial s}, \frac{\partial g_1}{\partial l_1}, \frac{\partial g_1}{\partial t_1} \text{ and } \frac{\partial g_2}{\partial s}, \frac{\partial g_2}{\partial l_2}, \frac{\partial g_2}{\partial t_2}.$$

For the consistency constraints $t_1 - A_1(s, l_1, t_2) = 0$ and

$t_2 - A_2(s, l_2, t_1) = 0$ we need

$$\frac{\partial A_1}{\partial s}, \frac{\partial A_1}{\partial l_1}, \frac{\partial A_1}{\partial t_2} \text{ and } \frac{\partial A_2}{\partial s}, \frac{\partial A_2}{\partial l_2}, \frac{\partial A_2}{\partial t_1}.$$

Example: Sensitivities in DAO vs FIO, cont.

Consider FIO:

$$\begin{aligned} \underset{s,l_1,l_2}{\textbf{minimize}} \quad & f(s, a_1(s,l_1,l_2), a_2(s,l_1,l_2)) \\ \textbf{subject to} \quad & g_0(s, l_1, a_1(s,l_1,l_2), a_2(s,l_1,l_2)) \geq 0 \\ & g_1(s, l_1, a_1(s,l_1,l_2)) \geq 0 \\ & g_2(s, l_2, a_2(s,l_1,l_2)) \geq 0, \end{aligned}$$

where a_1 and a_2 are computed in MDA

$$\begin{aligned} a_1 &= A_1(s, l_1, a_2) \\ a_2 &= A_2(s, l_2, a_1) \end{aligned}$$

Example: Sensitivities in DAO vs FIO, cont.

In FIO approach, we need to compute the sensitivities of the objective

$$f_{FIO}(s, l_1, l_2) = f(s, a_1(s, l_1, l_2), a_2(s, l_1, l_2)).$$

By the chain rule,

$$\begin{aligned}
\frac{\partial f_{FIO}}{\partial s} &= \frac{\partial f}{\partial s} + \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial s} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial s} \\
\frac{\partial f_{FIO}}{\partial l_1} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_1} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_1} \\
\frac{\partial f_{FIO}}{\partial l_2} &= \frac{\partial f}{\partial a_1}\frac{\partial a_1}{\partial l_2} + \frac{\partial f}{\partial a_2}\frac{\partial a_2}{\partial l_2}
\end{aligned}$$

We compute the derivatives of a_1 and a_2 by implicit differentiation of the multidisciplinary analysis equations

$$\begin{aligned}
a_1 - A_1(s, l_1, a_2) &= 0 \\
a_2 - A_2(s, l_2, a_1) &= 0
\end{aligned}$$

This yields

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial s} \\ \dfrac{\partial a_2}{\partial s} \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial A_1}{\partial s} \\ \dfrac{\partial A_2}{\partial s} \end{pmatrix},$$

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial l_1} \\ \dfrac{\partial a_2}{\partial l_1} \end{pmatrix} = - \begin{pmatrix} \dfrac{\partial A_1}{\partial l_1} \\ 0 \end{pmatrix},$$

and

$$\begin{pmatrix} I & -\dfrac{\partial A_1}{\partial a_2} \\ -\dfrac{\partial A_2}{\partial a_1} & I \end{pmatrix} \begin{pmatrix} \dfrac{\partial a_1}{\partial l_2} \\ \dfrac{\partial a_2}{\partial l_2} \end{pmatrix} = - \begin{pmatrix} 0 \\ \dfrac{\partial A_2}{\partial l_2} \end{pmatrix}$$

to be solved for the sensitivities of a_1 and a_2 wrt (s, l_1, l_2). (Referred to as the "generalized sensitivity equations" by Sobieski, 1990)

Autonomy of implementation

- The same elements are needed for sensitivities in SAND, DAO, FIO
- Can implement constituent elements with autonomy if do not integrate MDA via fixed-point iteration early
- The elements are integrated differently in FIO and DAO
- Analogous results for CO and OLD
- In principle, can re-arrange computational components associated with one formulation and obtain components for another
- Re-arrangement may require substantial effort
- For some formulations, the re-arrangement is straightforward
- May reformulate or use hybrid approaches (far vs. near solution)

n.alexandrov@larc.nasa.gov

Example: DAO vs FIO vs SAND (analysis and coupling constraints only)

Simplified FIO formulation: $\underset{x}{\text{minimize}} \quad f_{FIO}(x) \equiv f(x, a_1(x), a_2(x)),$

where, given x, we solve the MDA

$$\begin{pmatrix} \tilde{A}_1(x) \\ \tilde{A}_2(x) \end{pmatrix} = \begin{pmatrix} a_1 - A_1(x, a_1(x), a_2(x)) \\ a_2 - A_2(x, a_1(x), a_2(x)) \end{pmatrix} = 0$$

Simplified SAND formulation:

$$\begin{array}{ll} \underset{x, a_1, a_2}{\text{minimize}} & f_{SAND}(x, a_1, a_2) \equiv f(x, a_1, a_2) \\ \text{subject to} & \tilde{A}_1(x, a_1, a_2) = 0 \\ & \tilde{A}_2(x, a_1, a_2) = 0 \end{array}$$

Simplified DAO formulation:

$$\begin{array}{ll} \underset{x, a_1, a_2, t_1, t_2}{\text{minimize}} & f_{DAO}(x, a_1, a_2) \\ \text{subject to} & t_1 - a_1(x, t_1, t_2) = 0 \\ & t_2 - a_2(x, t_1, t_2) = 0 \end{array}$$

n.alexandrov@larc.nasa.gov

Example: DAO vs FIO vs SAND, cont.

W_i — basis of the null-space associated with the derivative of the block A_i. Relying on implicit differentiation and the derivations by Lewis, 1997, note the relationship among the sensitivities for the three methods:

- Suppose, (x, a) is feasible with respect to MDA. Then the (projected) gradients at (x, a) of FIO and SAND are related by

$$\nabla_x f_{FIO}(x) = W_{SAND}^T(x, a)\nabla_{x,a} f_{SAND}(x, a),$$

  where W_{SAND} denotes a particular basis for the null-space of $\nabla \tilde{A}^T$ in the SAND approach.

- Suppose that (x, a) is feasible with respect to MDA. Then

$$W_{DAO}^T \nabla_{x,a} f_{DAO}(x, a) = W_{SAND}^T(x, a)\nabla_{x,a} f_{SAND}(x, a)$$

Can use these relationships to implement a reduced-basis optimization algorithm for the three formulations with minimal modifications.

Sketch of a conceptual algorithm

Consider one step of a reduced-basis algorithm for the SAND formulation:

1. **Construct a local model of the Lagrangian about the current design.**
2. **Take a substep to improve feasibility.**
3. **Subject to improved feasibility, take a substep to improve optimality.**
4. **Set the total step to the sum of the substeps, evaluate and update.**

- **MDA after step 4 $\Longrightarrow$ a corresponding algorithm for FIO.**
- **Solving the disciplinary equations as in DAO $\Longrightarrow$ an algorithm for DAO.**
- **Passing between algorithms for distinct formulations is a straightforward step.**

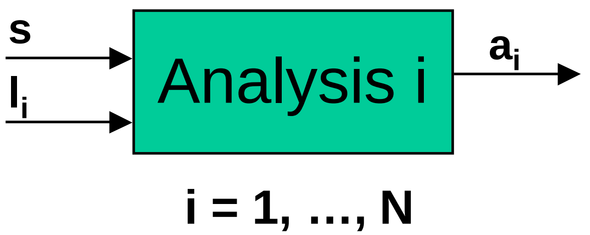
Our Currently Favorite Formulation: Expanded DAO

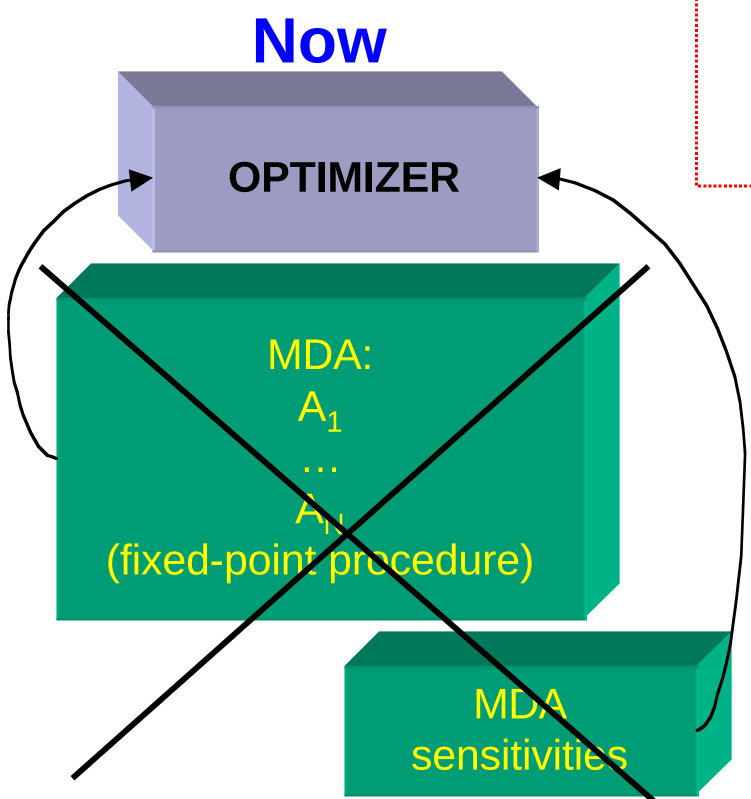
$$
\begin{aligned}
\underset{s,\sigma_0,\sigma_1,\sigma_2,l_1,l_2,t_1,t_2}{\textbf{minimize}} \quad & f_{DAO}(s, t_1, t_2) \\
\textbf{subject to} \quad & g_0(\sigma_0, t_1, t_2) \geq 0 \\
& g_1(\sigma_1, l_1, t_1) \geq 0 \\
& g_2(\sigma_2, l_2, t_2) \geq 0 \\
& t_1 = a_1(\sigma_1, l_1, t_2) \\
& t_2 = a_2(\sigma_2, l_2, t_1) \\
& \sigma_0 = s \\
& \sigma_1 = s \\
& \sigma_2 = s
\end{aligned}
$$

- **Expand variable space to relax the requirement that the disciplinary design constraints be satisfied with the system-level values of s**
- **Implementation autonomy, no MDA**
- **Single-level optimization problem - readily soluble**

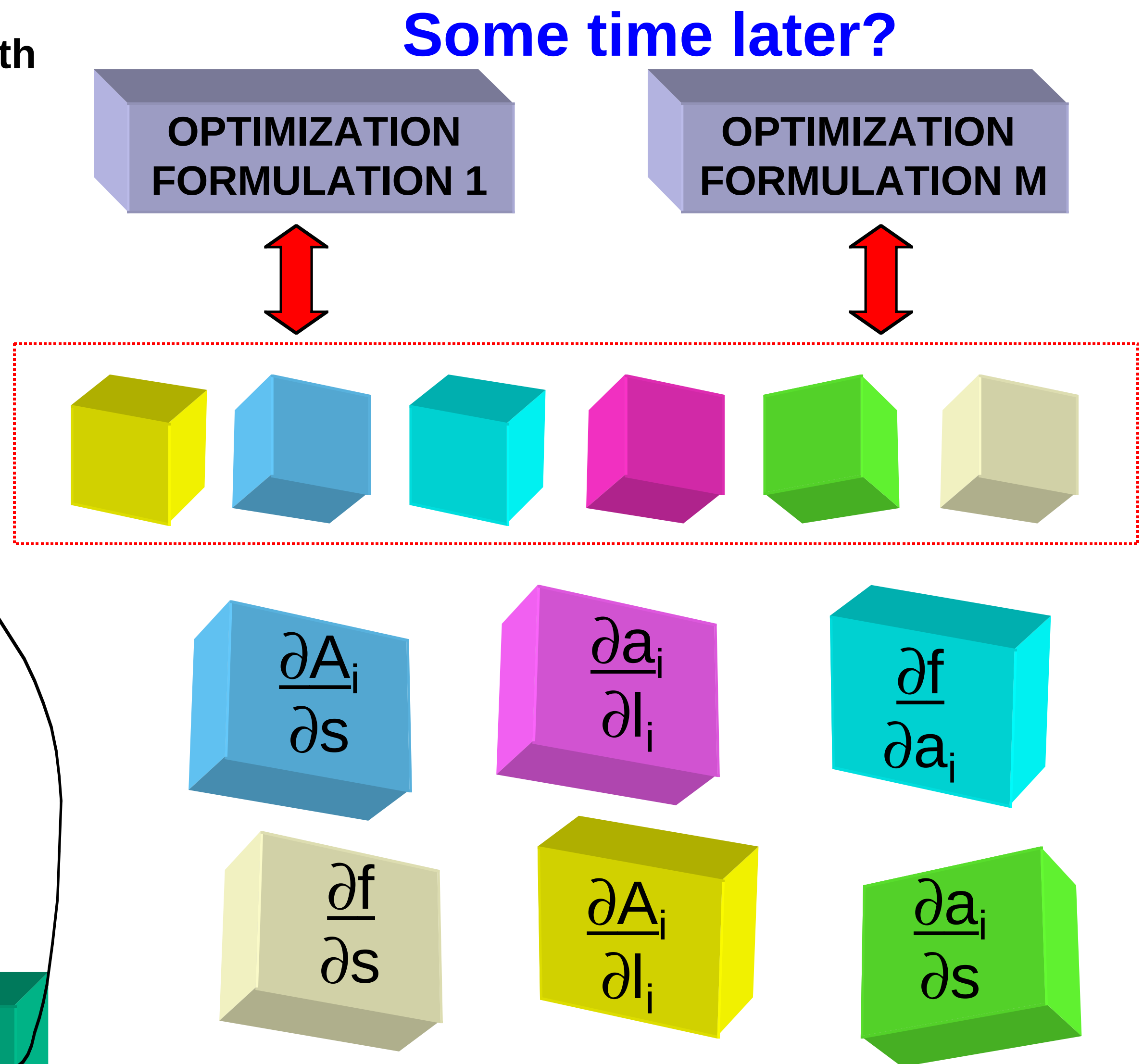
MDO Problem Synthesis / Implementation

Problem: design for objective f with





Laborious, expensive, one-time integration, difficult to transform/ expand



Expend the effort at the outset to implement analysis and sensitivity modules; easy to transform and expand: an opportunity for a general framework

n.alexandrov@larc.nasa.gov

Concluding Remarks

- Problem formulation is one of the deciding factors in practical solubility of the problem
- No single formulation is ideal for all problems
- Disciplinary function and derivative modules can ease implementation and enable some degree of disciplinary autonomy and dynamic re-configuration of the problem
- However…
  - There is a good reason for periodic reappearance of CO-like methods: handling of local design variables in disciplines is desirable
  - Unsolved problem: efficient, robust, method with full disciplinary autonomy
- Some other limiting factors in MDO and simulation-based optimization:
  - Extreme expense of function evaluations (addressed Wed.)
  - Insufficiently developed models